A Neural Network with Evolutionary Algorithm Implementation that Classifies Horizontal and Vertical Control Types by User search Performance

Weiliang Cao

Research School of Computer Science, Australian National University

{u6660031 }@anu.edu.au

Abstract. When using the search engine, the control type may influence users' efficiency, satisfaction and accuracy of searching. Aiming at the difference between horizontal and vertical control type, one three-layer neural network was designed to classify the two control types. To improve the performance of the model, techniques such as decrypting neural network data, evolutionary algorithm and network reduction are used. The accuracy of the final classification can reach up to 80%, which fully proves the influence of the two control modes on users' usage experience.

Keywords: Neural network, loss, classification, hidden neuron, horizontal control type, sigmoid, Relu, reduction, adjust, layer.

1 Introduction

The search engine on mobile devices is widely used by people every day, and its efficiency is worth studying. Its better performance can greatly save users' time and improve the search efficiency. At present, most search engines on mobile devices use vertical scroll control mode [1]. The database I chose studied whether using horizontal control mode can get the same or better results than using vertical control mode.

1.1 Survey in Database

The study on Pagination versus Scrolling in Mobile Web Search used professional equipment (They use a stationary phone with a camera and place sensors on the screen to record time) and designed reasonable tasks (Divide the search engine into two pages with three target locations per page. Let users target six locations for a search experience). Users are randomly selected and do not know the purpose of the experiment. The time, accuracy and satisfaction of the search were recorded. It is found that the performance of horizontal control in mobile search engines is as good as or even better than that of vertical control mode [1].

1.2 Modelled Question

In Pagination versus Scrolling in Mobile Web Search, vertical and horizontal controls differ in terms of task completion. Such as data of average first click time, correct click time and total time on SERPs, which showed horizontal control type had a better performance (cost less time). Additionally, horizontal control has higher accuracy and fewer rolls (The number of times scroll while using the search engine).

Then we can use the differences found above to establish a neural network. Use time to first click, time to right click, total time on SERPs, task completion duration, accuracy, satisfaction, scroll and target position as input. The target is classifying the vertical and horizontal control modes. So, use control type as output. This is a classification model

1.3 Outline of the Investigations

First of all, the dataset was preprocessed and split into training set and testing set. Secondly, a neural network was established, which take the attributes in the above investigation in terms of vertical and horizontal control modes as input, sets the vertical and horizontal control modes as label. Then, used the decrypting neural network data technology, the dataset was further analyzed and encoded. Then adjusted the hyperparameter (such as the number of neural network layers, the number of neurons in each layer) and the type of loss function and optimizer used. In the process of adjusting the number of neurons in hidden layers, successively used Network Reduction Technique and evolutionary algorithm.

In order to evaluate the performance of neural network, three methods were used, including the percentage accuracy, the intuitive representation accuracy of confusion matrix and the average accuracy calculated by cross validation.

2. Method

According to the findings in Pagination versus Scrolling in Mobile Web Search, respondents' performance varies between vertical and horizontal controls.

Figure 1[1]

Obviously, in the above four attributes reflecting the speed of the respondents' use of search engine, the time of the horizontal control is significantly less than that of the vertical control.

Target position	Type	Time to first click	Time to right click	Total time on SERPs	Task completion duration	Accuracy	Satisfaction	Scroll
All	Horizontal	29.042	39.844	35.230	61.445	0.778	5.542	0.625
	Vertical	32.983	46.997	40.610	68.254	0.701	5.590	0.667
Front	Horizontal	26.418	36.751	32.447	58.661	0.792	5.597	0.417
	Vertical	33.073	40.862	37.366	63.654	0.819	5.736	0.542
Back	Horizontal	31.666	42.938	38.014	64.229	0.764	5.486	0.833
	Vertical	32.893	53.132	43.855	72.855	0.583	5.444	0.792

Form [1]

The above table shows averages of all the attributes (front and back represent the positions of the previous and last pages, respectively). Several other characteristics can be seen, that is, the subjects use the horizontal control mode to be more accurate, less times of sliding, but the satisfaction is basically the same as the vertical control mode.

Consequently, I set up a simple two-layer neural network by taking the columns in the table other than control type as input and set control type as output. Tried different optimizers and loss functions. Successively used decrypting neural network data technique, network reduction technique and evolutionary algorithm.

2.1 Model structure

The input is a list of participants' performance, including the current test's target location (1,3,5,6,8,10), the subject's first click time, the correct click time, the total time and delay in the search engine page, as well as accuracy and satisfaction, and the number of slides.

1. Load and preprocessing raw data
2. Input encoding
3. Output encoding
4. Hidden Layer processing
5. Predicting a class

The neural network consists of four layers, including an input layer, two hidden layers and an output layer. Among them, the sigmoid function is used as the activation function of the hidden layer and the output layer.

In the process of training, the technique of encoding input and output is used, which recodes some data with unclear fuzzy rules, improves the accuracy of the data, and enhances the fit degree of some scattered data to the neural network.

Three methods were used to verify the performance of the model, including direct calculation of accuracy, calculation of confusion matrix, and the use of cross validation techniques.

Finally, this model has a good performance, which is a big improvement over the simple two-layer neural network that was not processed with any technique at the beginning. It went from 53 percent accuracy to 75 to 83 percent accuracy which is a significant improvement. The detailed process and discussion of the analysis of the data will be presented in the following sections.

2.2 decrypting neural network data

2.2.1 data preprocessing and input encoding

The selection of the appropriate data pattern is crucial to the model. The application of input coding should first analyze the data and select the appropriate data type [2]. The right data type can better express the meaning of data in the natural world. It is found that the data that reflects the user's search time and satisfaction is in the form of specific Numbers, which can well reflect the relationship between each other. No coding is required. Therefore, we simply preprocessed the data, including subtracting the average value from the data set and enlarging the range of satisfaction to make the difference between the data larger.

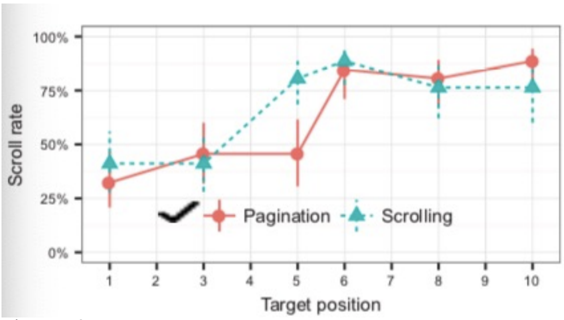

Figure 2

For the target position, which have obvious difference based on paging. The first three positions are similar in performance, and the last three positions are similar. The current code is 1,3,5,6,8,10. The numerical distance between the positions 5 and 6 is less than that between the positions 3 and 5, which obviously does not correspond to the fact that they exist in the natural world. Consequently, I'm going to recode the target position into two properties: 'target position' and 'target position2'. Positions 1, 3, 5 are coded as' 1 'and' 0 ', and positions 6, 8 and 10 are coded as' 0 'and' 1 '. This coding can provide more accurate and efficient input to the classification model.

2.2.2 Output Encoding Technique

This neural network is a classifier, the application of output coding technique, the output needs to be converted into a suitable form. Appropriate output coding can train all the output neurons in the corresponding training path to improve the performance of the model [3]. In this model, fortunately, we only need to encode both vertical and horizontal control modes of output. First of all, we can change the vertical code to 1 and the horizontal code to 0.

Since there are only two output neurons, in this case all output neurons are programmed into the computation process. Consequently, equilateral coding technique is not needed (When multiple outputs are used to represent different categories, each output has a corresponding value to represent each category).

2.2.3 Adding Random Noise Technique

This method is added at the end. The addition of gaussian noise can improve the stability of the training model [4]. After the mean value and variance of the noise are defined, the gauss function is used to automatically generate the noise and add it into the data. This can effectively avoid overfitting and increase the accuracy of the model.

2.2.4 Pattern Reduction Technique

To remove abnormal model can significantly improve computing speed [2]. in data preprocessing phase, when calculate the error of web browsing's average value, there shows a not small number, while most value in this property should be 0. Obviously, individual points with big value exist. Delete them could improve the data for using. Deleting them also reduces the size of the data. However, because the data set itself is not big and the classification feature is not particularly obvious, the deletion of data leads to the decline of the accuracy.

2.3 Hidden Layers and Activation Function

On the question of choosing the number of hidden layers, first of all, you need at least a hidden layer to the fitting of nonlinear separable function (that is, from the performance of the respondents to use what kind of control mode), the second hidden layer can make any decision boundary of the model to represent arbitrary precision, theoretically more hidden layers can learn the description of the more complex, but at the same time also can bring a overfitting problem and cost too much time. After an attempt to set 1, 2, 3, and 4 hidden layers, two hidden layers were found to be the most accurate and efficient.

After comparing ReLU function, tanh function and sigmoid function, it is found that this neural network model mainly solves the classification problem, so it is particularly suitable to use sigmoid function, which maps the values to the interval (0,1), which can solve the nonlinear mapping problem well [5].

sigmoid function :

$$g(z) = \frac{1}{1+e^{-z}}$$

(z is the input of the function)

2.4 Cross Validation and Confusion Matrix

In order to verify the results of the model, a method of cross-validation was adopted, call the Kfold function in the library, use k=20 which divided the data into 20 parts randomly, make one part as the test set and the other 19 parts as the training set in turn, and then average the accuracy. The results are relatively accurate, avoiding coincidences and increasing the number of training sets.

By using the obfuscation matrix, the comparison results of the real value and the predicted value can be reflected intuitively [6]. So as to understand the results more quickly and provide relevant Suggestions for the modification of the model.

2.5 Optimizer and Loss Function

Adam is used as optimizer, It combines the advantages of Adagrad being good at dealing with sparse gradients and RMSprop being good at dealing with non-stationary targets. The advantage of Adam is that after bias correction, the learning rate of each iteration has a certain range, which makes the parameters relatively stable.

Finally, CrossEntropyLoss, were selected. CrossEntropyLoss can gradually increase the cross entropy as the predicted probability deviates from the actual target, which combines nn.logsoftmax () and nn.Nllloss () in a separate class. Consequently, you don't need to add softmax to network to use it. It has optional parameter weights, 1dimention Tensor, which assign weights to each class. It's an excellent way to solve this classification problem.

2.6 Network Reduction Technique

Many of the output of the hidden neuron is repetitive, which will affect the computational efficiency and accuracy of the neural network [8]. The output of the hidden neuron is output as a vector, and the middle Angle of each vector is calculated by using the mathematical formula. If it is less than 15 degrees or greater than 165 degrees, then count+1, the average count value is found out three times in the loop, and the count value is subtracted from the corresponding hidden layer. However, the final result shows that this method is not applicable to this neural network model. Maybe due to the limited data input, the accuracy does not improve after a round of hidden neuron deletion. After comparing the hidden neurons with this method, we found that there were still duplicate neurons, and the number of hidden neurons would be too small after deleting again, and the accuracy would not be significantly improved.

2.7 Evolutionary Algorithm

In 2.6, the reduction of the number of neurons in the hidden layer did not improve the accuracy, but it was a factor that greatly affected the accuracy. Therefore, I applied the evolutionary algorithm to find the appropriate number of neurons in the hidden layer. This will optimize the neural network model to some extent.

Basically setting: First, I set the neural network with different numbers of two hidden layers as individuals in the population. The number of neurons in the first and second hidden layers was set to the x and y chromosomes. The method of crossover is one-point crossover (when crossover, the decimal number is converted into a binary number), randomly select one point in the parents' chromosomes and get different part from mother and father. I set the mutation rate to random. I set the accuracy of the neural network as the fitness function. The value of accuracy is the health value. Parents are selected based on their health values

Training process:
1. initialize first generation (randomly choose in the range)
2. use fitness function to get the health values of every individuals
3. calculate the possibility for chosen based on fitness

use possibility to get parents to crossover and mutation to get next generation (save the best one as elitism, do not change it, directly save to next generation)
4. save new generation and repeat step 2-4
5. save every elitisms' chromosomes and fitness value to report excel document and plot them
Finally, we can find lots of good individuals with high fitness values, which could be chosen for comparison.

3 Results and Discussion

After applying decrypting neural network data Technique, Pattern Reduction Technique and evolutionary algorithm, the performance of the model was greatly improved, from the initial around 50% (simple one-layer neural network) to the final 65-83% (Due to the small amount of data in the test set, there will be a large fluctuation in accuracy). This section describes the final model setup and discusses the enhancement process.

3.1 Model Setup and Final Result

The dataset is spilt into training part and testing part, randomly select 95% data as training data and 5% data as testing data. Training data is used to train the neural network for 800 loops and then use testing data to obtain the final result. At first, for a simple comparison, choose loss function, optimizer, and the number of hidden layers. Then use evolutionary algorithm to find suitable number of neurons in hidden layers, every generation has a size of 5, the max generation is 60. The following figure shows the different accuracy distributions corresponding to the number of neurons in different hidden layers. This suggests that it is important to select the right number of neurons in the hidden layers.

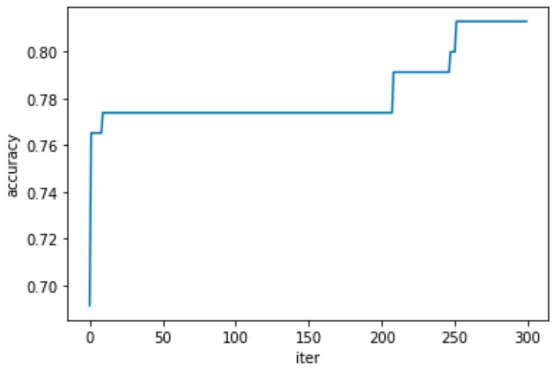

Figure 3

	x1	x2	accuracy
0	85	51	0.691304
1	97	50	0.765217
2	108	34	0.752174
3	35	71	0.691304
4	106	101	0.669565
5	67	93	0.717391

This chart is part of the report generated during training which store the chromosomes of individuals in population. This is an intuitive way to pick the right number of neurons in hidden layers. Finally, I picked 200, 20 as they appear several time in the process of training evolutionary algorithm, and the accuracies of them are very high.

The optimal setting of this model is 9 input neurons, including 8 data attributes (time to first click, time to right click, total time on SERPs, task completion duration, accuracy, satisfaction, scroll and target position) and one input that adds all time data, two hidden layers, 165 neurons in the first layer, 20 neurons in the second layer, and two output neurons. The activation functions of both the hidden layer and the output layer use the sigmoid function. Optimizer uses Adam function, and Loss function uses CrossEntropyLoss.The input target position and output are encoded. Get rid of outliner. The neural network performance results were evaluated using cross-validation function and confusion matrix.

Result example (due to small test dataset, the result fluctuates a lot)：

Testing Accuracy: 80.00 %

Confusion matrix	horizontal	vertical
horizontal	8	0
vertical	3	4

Cross validation result:

time	1	2	3	4	5	6	7	8	9
Accuracy	61.23%	80.00 %	71.43 %	71.43 %	62.33%	85.71 %	59.72%	61.12%	71.73%
time	10	11	12	13	14	15	16	17	18
Accuracy	71.43%	64.81%	55.62%	62.51%	54.62%	72.65%	59.12%	62.62%	82.86 %
time	19	20	total						
Accuracy	71.43%	72.86 %	67.76%						

3.2 Discussion of Enhancement Process

At first, I established a normal one layer network with accuracy of around 51%, which is pretty low. The most significantly improve appear after I adjust the number of hidden layers, loss function and activation function. The accuracy reached to around 63%, with 2 hidden layers, sigmoid function as activation function and CrossEntropyLoss as loss function. As I introduced above, more intermediate neurons can better transmit the learned data features, the appropriate activation function can better realize the classification (Mapping to the appropriate interval facilitates classification), and the correct loss function can better calculate the classification loss. Then I used decrypting neural network data technique which include input and output encode, adding noise, pattern reduction. This slightly increased the accuracy. More suitable data could provide more important features. Then network deduction technique is used. Surprisingly, there is no improvement and sometimes may cause decrease of accuracy. This is cause by the feature itself is not obvious and the original number of hidden neurons are not suitable. Finally, Evolutionary algorithm significantly improved the performance of the model. The accuracy reached to 65% and above. Right selection of numbers of hidden neurons could help transfer more features among the neural network.

4 Conclusion and Future Work

A three-layers neural network model is designed to classify the vertical and horizontal control types with users' performance data. A good classification of the control type results justifies the discovery in Pagination versus Scrolling in Mobile Web Search and illustrate the potential development value of the horizontal control type. In the training process of this model, decrypting neural network data technique and evolutionary algorithm are adopted, the input and output data of the neural network are adjusted reasonably, and the number of neurons in the hidden layer is explored by genetic algorithm. The implementation of these techniques improved the performance of the model.

In future work, we can implement the evolutionary algorithm to optimize the weight of the neural network, which would improve the performance of the model more directly. Try other crossover and selection methods, such as a hall of fame. In addition, the number of search engine interfaces can be increased when investigating respondents. If the it can increase to three or four pages, the collected data will more easily prove the advantages of horizontal control type.

References

1. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., & Yoon, H. J. (2016, October). Pagination versus scrolling in mobile web search. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 751-760).
2. Bustos, R. A., & Gedeon, T. D. (1995). Decrypting neural network data: a GIS case study. In Artificial Neural Nets and Genetic Algorithms (pp. 231-234). Springer, Vienna.
3. Guiver, J. P., & Klimasauskas, C. C. (1991). Applying neural networks, Part IV: improving performance. PC AI Magazine, 5, 34-41.
4. Da Silva, I. B. V., & Adeodato, P. J. (2011, July). PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets. In The 2011 International Joint Conference on Neural Networks (pp. 2664-2669). IEEE.
5. Lin, C. W., & Wang, J. S. (2008, May). A digital circuit design of hyperbolic tangent sigmoid function for neural networks. In 2008 IEEE International Symposium on Circuits and Systems (pp. 856-859). IEEE.
6. Lewis, H. G., & Brown, M. (2001). A generalized confusion matrix for assessing area estimates from remotely sensed data. International journal of remote sensing, 22(16), 3223-3235.
7. Bottou, L. (2012). Stochastic gradient descent tricks. In Neural networks: Tricks of the trade (pp. 421-436). Springer, Berlin, Heidelberg.
8. Gedeon, T. D., & Harris, D. (1991). Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications (Vol. 1, pp. 119-126).